Kaggle Regression Problem

Arlo Encarnacion, Mike Pastuovic, Rishab Jayanthi

Northwestern

Train and Test Data

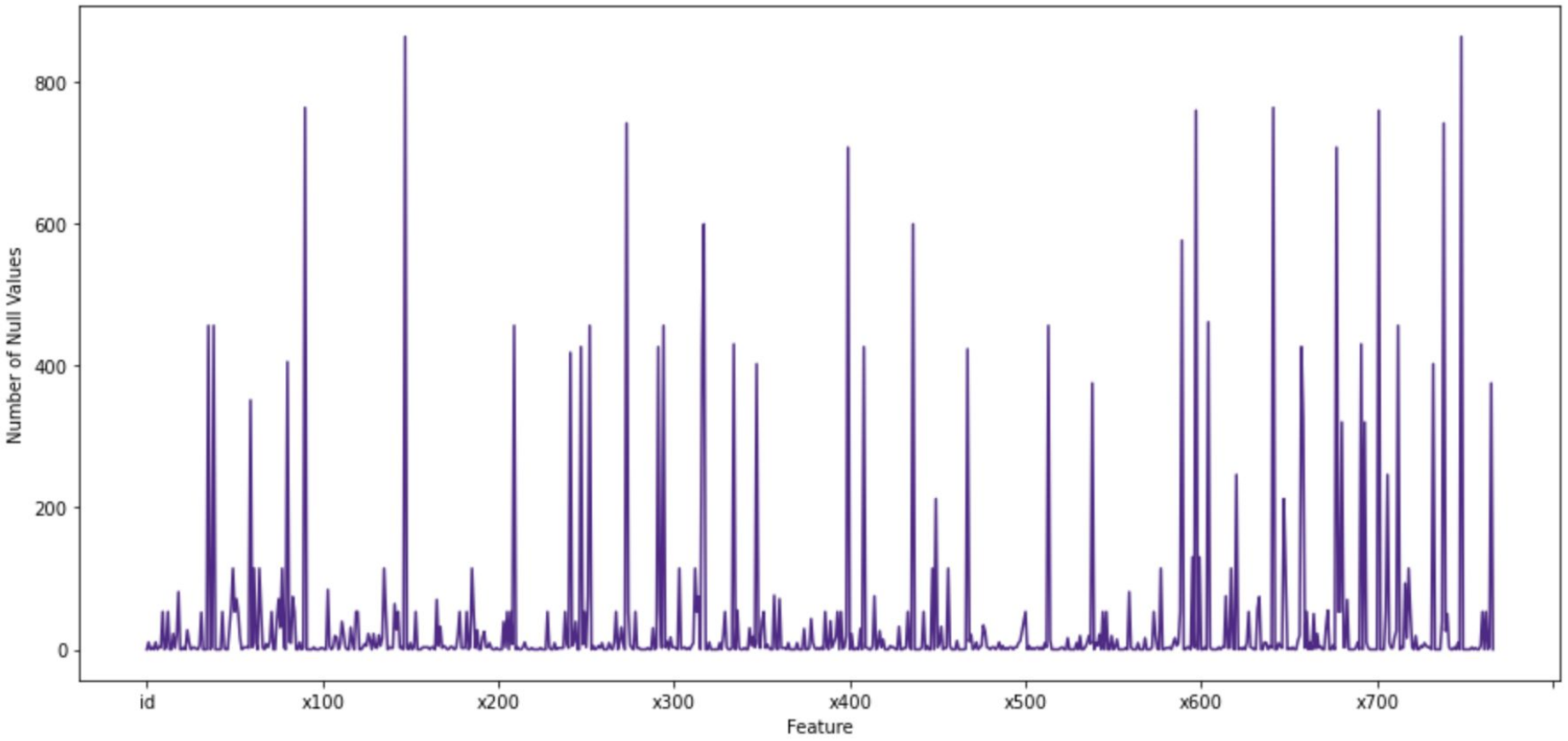
	id	x001	x002	x003	x004	x005	x006	x007	x008	x009	...	x757	x758	x759	x760
0	0	9.681860e+10	6991.15	7.76	0.00380	5.378811e+09	0.31	266117.20	934577.0	14539.0	...	0.0007	297281012	0.13	5.0
1	1	3.304810e+09	13914.43	5.37	0.00015	1.652405e+09	0.00	11927742.92	1798051.0	1051272.0	...	0.1136	3320000000000	0.08	661.0
2	2	3.218944e+10	3991.98	5.77	0.00010	2.476111e+09	0.00	774385.01	375738.0	144143.0	...	0.0029	100474819	0.39	39.0
3	3	1.288000e+10	15937.45	5.86	0.00020	2.146667e+09	0.00	6324375.16	1932094.0	10055.0	...	0.0000	348000000000	0.25	2.0
4	4	3.063412e+10	3621.00	7.52	0.00060	1.392460e+09	0.21	169860.29	474253.0	17914.0	...	0.0005	109546590	0.11	11.0
5	5	4.377769e+10	27776.26	6.02	0.00505	5.472212e+09	0.50	10797026.17	4501083.0	7538720.0	...	0.6223	154000000000000	0.52	1883.0
6	6	1.282546e+10	6215.45	6.07	0.00040	1.832208e+09	0.35	1509434.16	780135.0	2408.0	...	0.0000	1760000000000	0.26	2.0
7	7	1.583740e+10	17060.72	5.73	0.00275	2.262486e+09	0.49	10151935.50	2262318.0	4887848.0	...	0.4630	7170000000000	0.86	2007.0
8	8	5.844890e+10	9878.51	5.51	0.00185	6.494322e+09	0.42	2935716.59	1370456.0	1206.0	...	0.0004	552000000000	0.01	3.0
9	9	2.329566e+10	25682.32	6.88	0.00030	1.791974e+09	0.27	2445584.89	3319395.0	33620.0	...	0.0003	3090000000000	0.62	6.0
10	10	5.519027e+10	1184.89	5.54	0.00075	4.599189e+09	0.28	125504.55	96796.0	42338.0	...	0.0026	188254307	0.96	26.0
11	11	1.440000e+11	5236.85	7.35	0.00275	5.151139e+09	0.26	335357.91	622837.0	32726.0	...	0.0001	34201871001	0.87	3.0
12	12	5.056351e+09	16790.80	5.73	0.00045	1.685450e+09	0.47	10046694.12	2240226.0	4842630.0	...	0.4708	775000000000	0.01	1974.0
13	13	7.704705e+10	23964.99	5.30	0.00405	5.503361e+09	0.45	12223295.21	3496555.0	6238.0	...	0.0008	76600000000000	0.86	5.0
14	14	3.658658e+10	9281.74	6.62	0.00020	5.226654e+09	0.00	1379595.19	1469295.0	16798.0	...	0.0001	229949280	0.02	1.0
15	15	4.708059e+10	34758.12	6.39	0.00200	1.569353e+09	0.30	3322926.09	3702933.0	915.0	...	0.0001	123941961	0.01	1.0
16	16	8.693267e+09	928.21	6.79	0.00035	2.897756e+09	0.47	60379.35	112326.0	14047.0	...	0.0001	602000000000	0.20	4.0
17	17	5.575942e+10	22681.50	6.14	0.00250	1.467353e+09	0.27	3960684.80	2697351.0	73031.0	...	0.0012	1	0.00	36.0
18	18	1.758548e+10	14436.69	7.09	0.00010	4.396371e+09	0.00	1420430.68	1831482.0	4280.0	...	0.0000	501000000000	0.02	0.0
19	19	2.290484e+10	3510.78	6.72	0.00125	3.817473e+09	0.33	265750.95	433945.0	18077.0	...	0.0004	12856806	0.01	6.0

Train shape: (5380,767)

Contains target value column

Test shape: (4403,766)

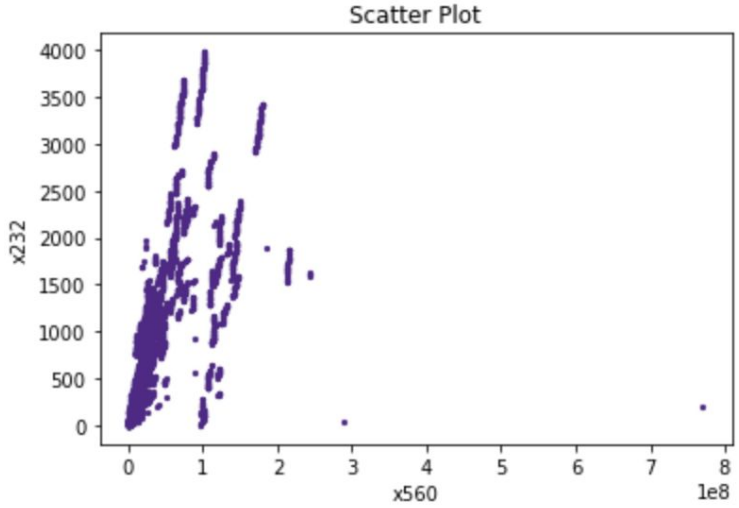
Null Values

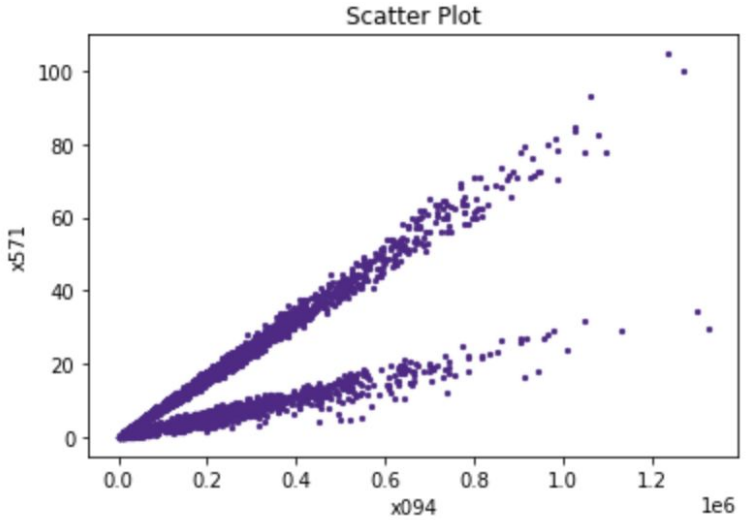

Number of Null Values
800
600
400
200
0
id
x100
x200
x300
x400
x500
x600
x700
Feature

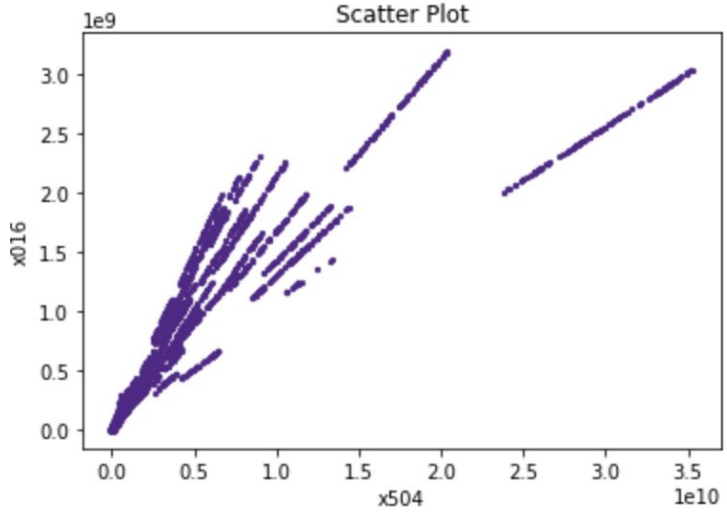
Correlation Matrix

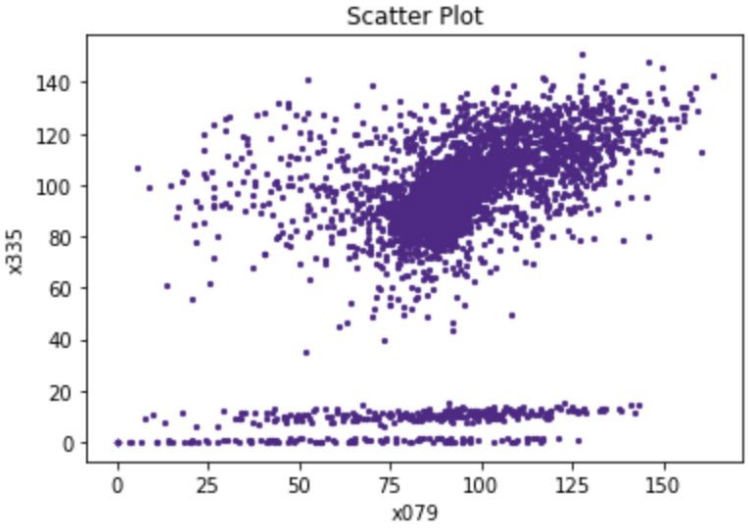
	id	x001	x002	x003	x004	x005	x006	x007	x008	x009	...	x757	x758	x759
id	1.000000	0.014652	-0.001859	0.001691	0.004353	0.024493	0.014469	-0.003433	-0.001495	-0.023534	...	-0.014413	0.009485	0.008319
x001	0.014652	1.000000	-0.102542	0.153749	0.379679	0.493197	0.247154	-0.165646	-0.103770	-0.032586	...	-0.052222	-0.006532	-0.016382
x002	-0.001859	-0.102542	1.000000	-0.177286	0.035330	-0.180123	0.071401	0.463602	0.961286	0.103345	...	0.058053	0.104158	0.045745
x003	0.001691	0.153749	-0.177286	1.000000	-0.066532	-0.052850	-0.113437	-0.500092	-0.197902	-0.179138	...	-0.252394	-0.137534	-0.091690
x004	0.004353	0.379679	0.035330	-0.066532	1.000000	0.231413	0.222151	0.126973	0.075906	0.038667	...	0.082101	0.021076	0.011411
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
x762	-0.018877	-0.058439	0.101158	-0.236631	0.046296	0.042812	0.098511	0.366232	0.150967	0.942280	...	0.842035	0.227901	0.042165
x763	-0.009983	0.629004	0.017972	0.204255	0.225874	-0.128443	0.226026	-0.154933	-0.019153	-0.085807	...	-0.100417	-0.086515	-0.037527
x764	-0.006646	-0.149572	0.819495	-0.500867	0.086455	-0.025749	0.116250	0.741603	0.853014	0.244453	...	0.250796	0.210006	0.094214
x765	0.001720	-0.005274	-0.022161	-0.005638	0.005585	0.060317	-0.014770	-0.011229	-0.012326	-0.003104	...	-0.002085	-0.001432	-0.032845
y	-0.018863	0.002661	-0.065831	0.201294	-0.021558	-0.060656	-0.122274	-0.109782	-0.074385	-0.050395	...	-0.059782	-0.107359	-0.117061

Correlation Plots









Multicollinearity

Correlation = 1.0

```
x206  x673    1.0
x495  x418    1.0
x470  x206    1.0
x179  x224    1.0
x539  x550    1.0
              ...
x546  x762    1.0
x547  x372    1.0
x046  x403    1.0
x372  x339    1.0
x471  x279    1.0
Length: 438, dtype: float64
```

Correlation > 0.7 and < 1.0

```
x055  x104    1.000000
x224  x104    1.000000
x104  x224    1.000000
      x055    1.000000
x179  x104    1.000000
                  ...
x232  x560    0.700164
x259  x560    0.700163
x560  x259    0.700163
x358  x216    0.700030
x216  x358    0.700030
Length: 26116, dtype: float64
```

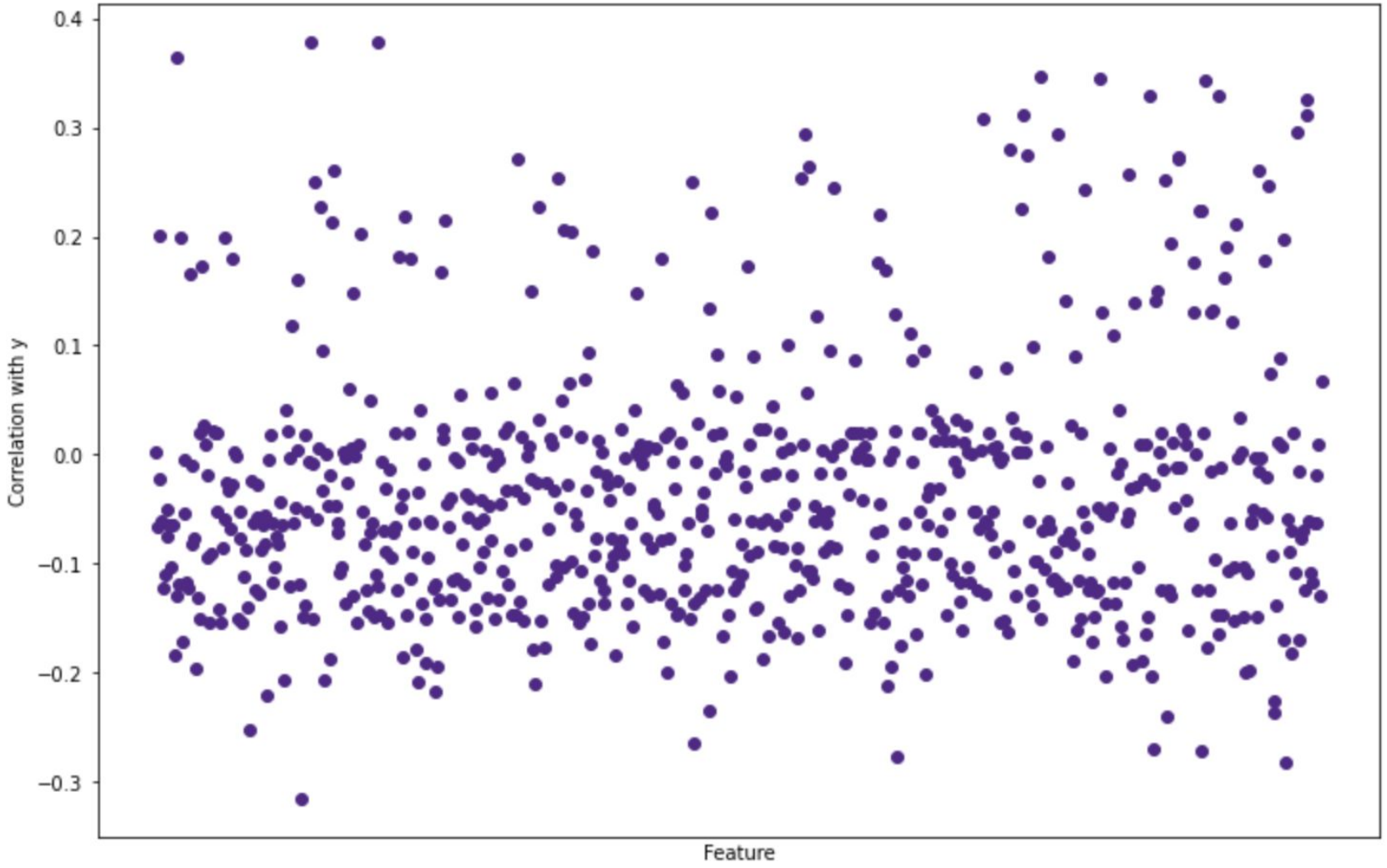
Multicollinearity

Filtered for feature combinations that yielded a correlation of 1.0 (absolute value)
Number of features that appeared: 113

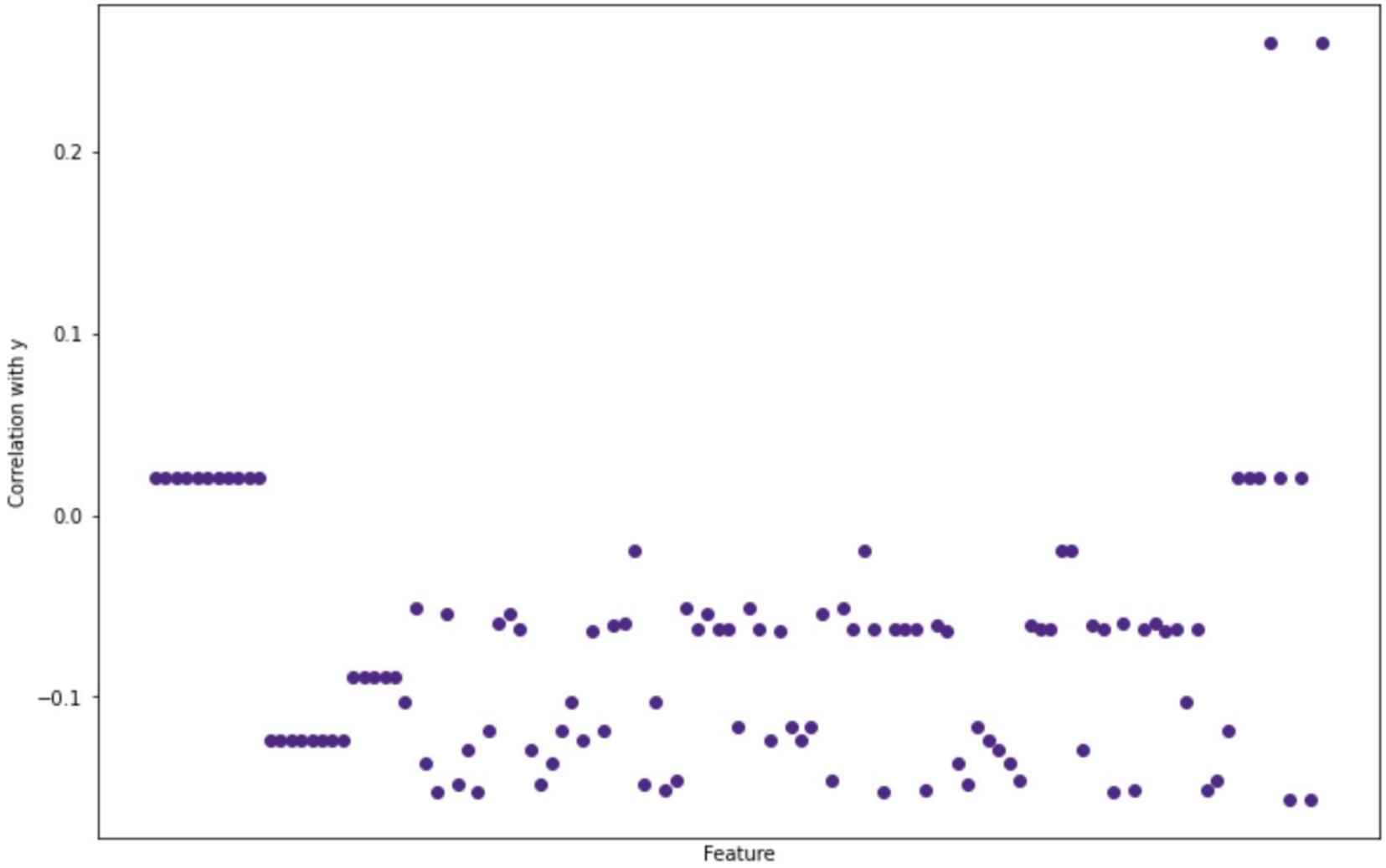
Filtered for feature combinations that yielded a correlation over 0.7 (absolute value) but not including 1.0
Number of features that appeared: 687

All features in the correlation of 1.0 list
also showed up in the > 0.7 list

Correlation with y



Correlation with y



Multicollinearity

Of the variables that display perfect correlation with others, all 113 of them were involved in at least two pairs of perfect correlation.

After removing those variables: 630 left

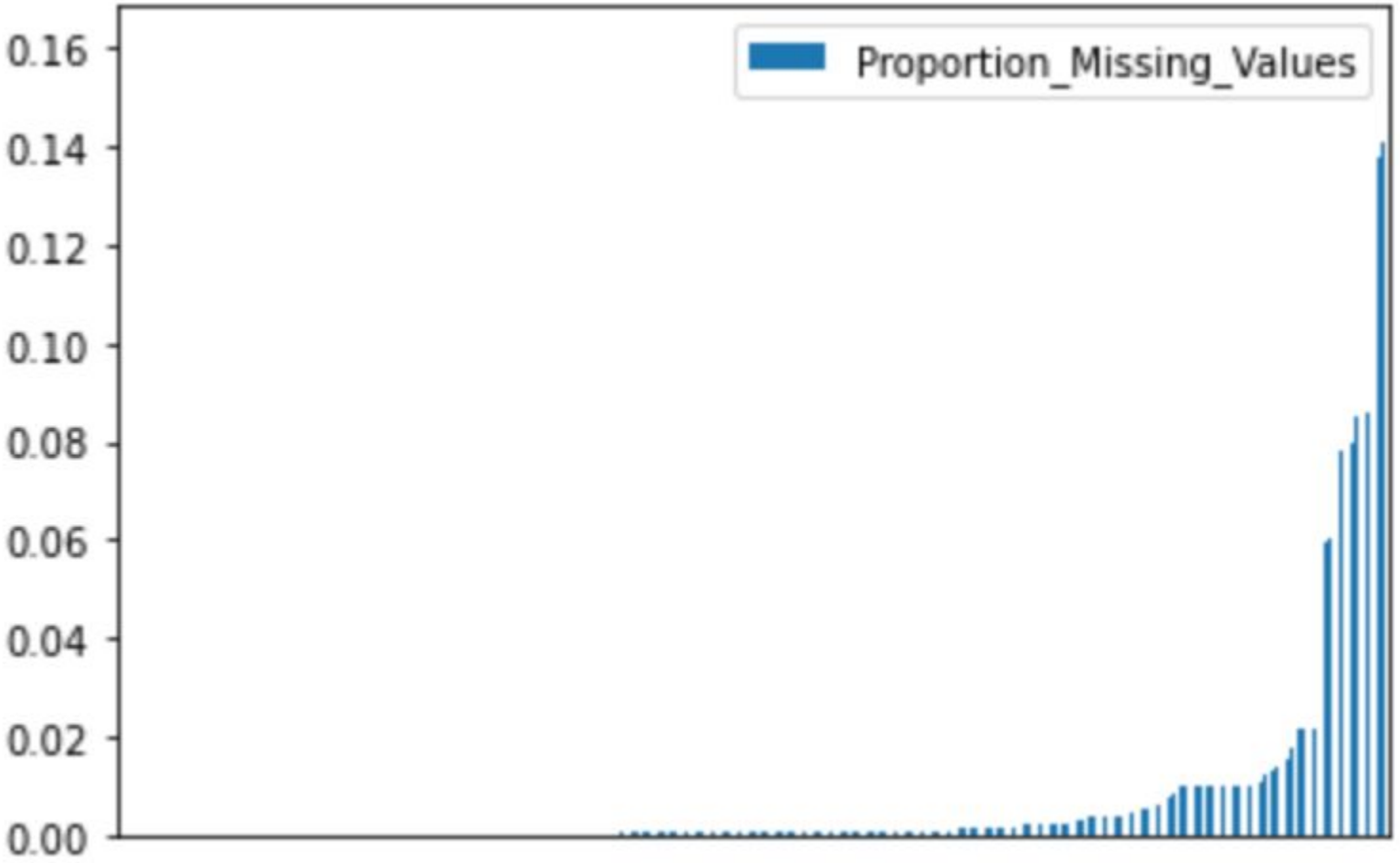
```
[55]: pd.set_option('display.max_rows', None)
      print(train1.corr()['y'].abs().sort_values(ascending = False))
```

```
y       1.000000
x724    0.260539
x118    0.260539
x279    0.152911
x471    0.152911
x479    0.152911
x058    0.152911
x179    0.151002
x224    0.151002
x352    0.151002
x055    0.151002
x615    0.148011
x650    0.148011
x144    0.148011
x715    0.148011
x455    0.146579
x377    0.146579
x237    0.146579
```

```
[57]: pd.set_option('display.max_rows', None)
      print(train.corr()['y'].abs().sort_values(ascending = False))
```

```
y       1.000000
x146    0.378696
x102    0.378436
x014    0.364737
x581    0.346549
x619    0.344101
x687    0.343842
x696    0.329630
x651    0.329630
x755    0.324916
x096    0.315185
x756    0.312263
x569    0.311497
x543    0.308728
x749    0.296195
x591    0.293075
x427    0.293073
x742    0.282411
x561    0.279915
x488    0.275807
x572    0.273892
x670    0.272551
x685    0.271773
x239    0.271436
x669    0.270390
x654    0.269544
x430    0.264170
x355    0.263551
x724    0.260539
x118    0.260539
x638    0.257546
x425    0.254060
x265    0.253899
x661    0.252327
x062    0.252143
x353    0.250863
x105    0.249884
x731    0.246480
x447    0.244871
```

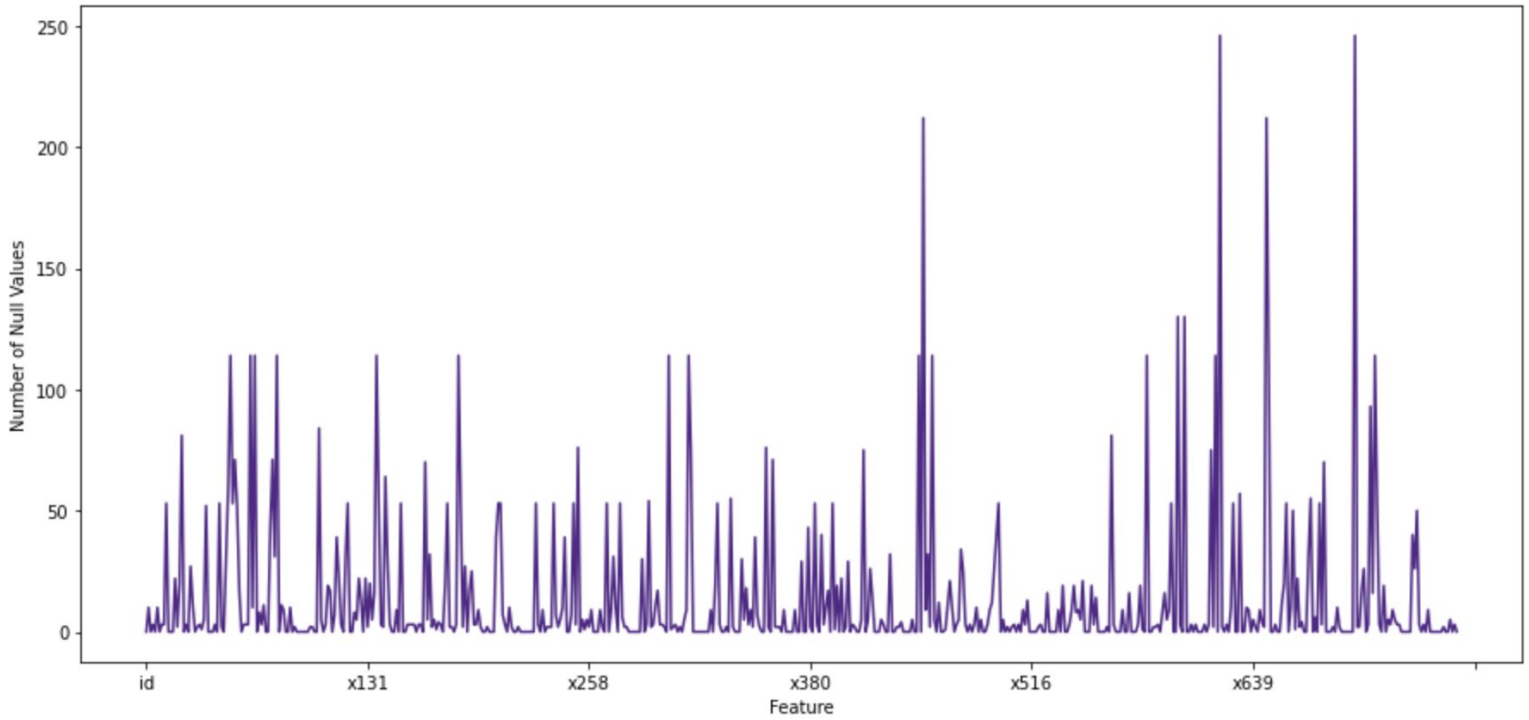
Null Values


Proportion_Missing_Values
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00

Null Values

We still have numerous columns with high numbers of null values.

We went ahead and removed columns where at least 5% of the data was null (~250 rows)



Imputation

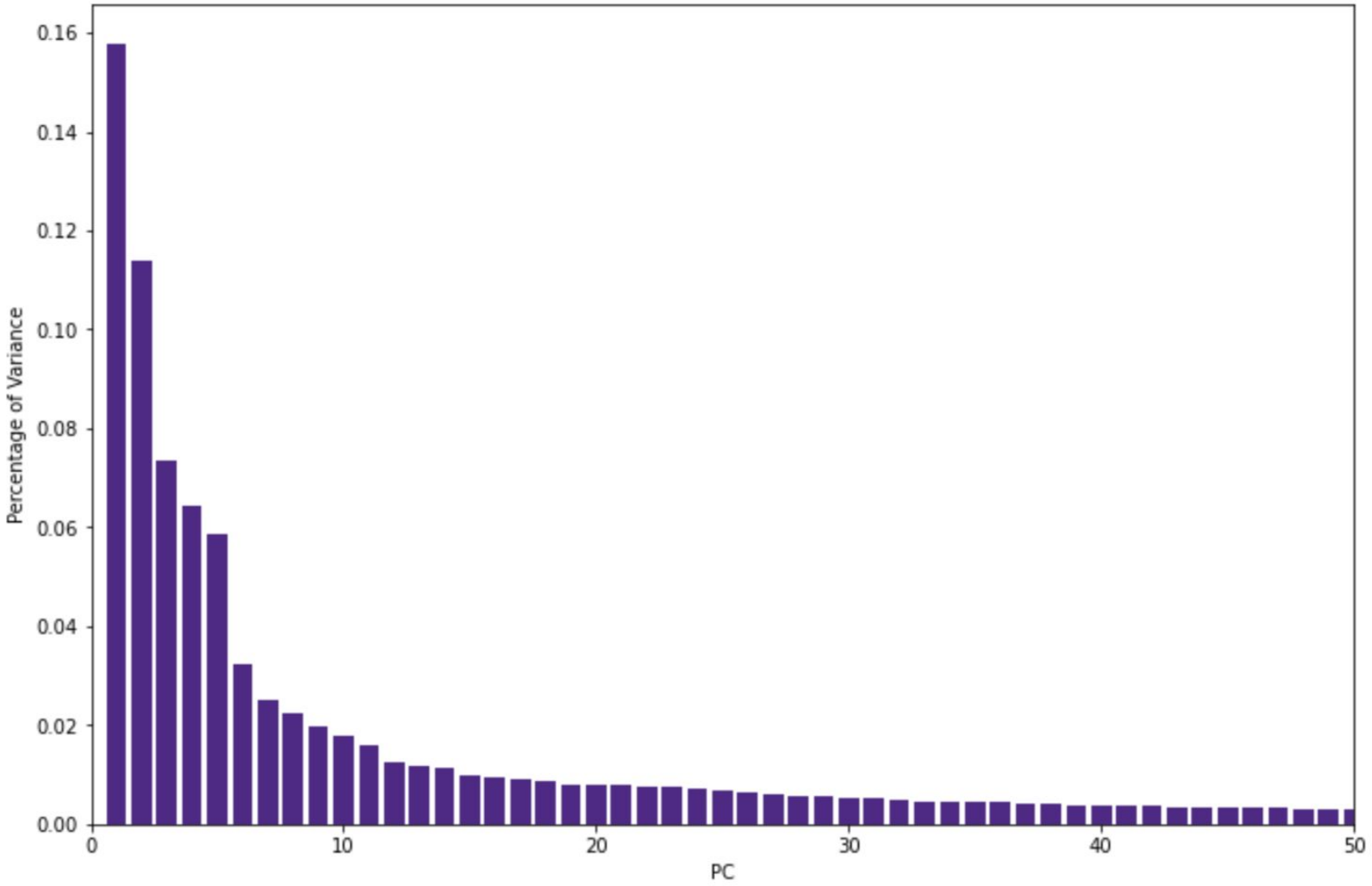
Imputing values for remaining columns with missing values

K-Nearest Neighbors Algorithm finds samples that are the closest in the training set, and takes an average of these points to impute the missing value

```
imputer = KNNImputer(n_neighbors=5)
X = pd.DataFrame(imputer.fit_transform(X),columns = X.columns)

Xtest_imp = pd.DataFrame(imputer.fit_transform(Xtest),columns = Xtest.columns)
```

PCA


0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Percentage of Variance
0
10
20
30
40
50
PC